2016 Fifth European Workshop on Software-Defined Networks

IoT ecosystem over programmable SDN infrastructure for Smart City applications

Łukasz Ogrodowczyk, Barosz Belter
Poznan Supercomputing and Networking Center (PSNC)
Poznań, Poland
Corresponding author: lukaszog@man.poznan.pl

Marc LeClerc
NoviFlow Inc.
Montreal, Canada
marc.leclerc@noviflow.com

***Abstract*—This demo paper presents an innovative Software Defined Networking (SDN) based approach to deploying Internet of Things (IoT) applications for Smart Cities. The Poznan Supercomputing and Networking Center (PSNC) together with NoviFlow Inc. and Spirent have jointly developed a demonstration showing how programmable SDN infrastructure can be utilized to significantly simplify the onboarding and provisioning of end-to-end IoT solutions for use in multi-tenant networks. The demo features a dynamic global view of the deployed IoT resources with their associated network connections, and the use of OpenFlow Experimenter-based extensions to trigger automated detection and onboarding of IoT devices, ("things") as well as the insertion of metadata into IoT device flows to automate service provisioning. The demonstration also features an SDN application that interfaces between the SDN controller and the cloud orchestrator to instantiate dedicated IoT services inside LXC light containers. The demonstration architecture includes a variety of typical IoT sensors as well as a Spirent TestCenter (STC) emulating metro scale IoT network workflows, interconnected to the cloud via a network composed of NoviFlow 2128 OpenFlow 1.3 switches. To illustrate the use of the solution in a real-world setting, the Poznan Smart City use case is presented, showing how a single common SDN-based platform can be utilized to "slice" a city into multiple smart spaces running over a shared network and cloud infrastructure, and how OpenFlow-enabled network infrastructure can be used to automate the deployment of IoT devices for use in multi-tenant, cloud-based applications.**

***Keywords*—IoT, SDN application, LXC, OpenFlow, SmartCity**

I. Introduction

Research in the area of IoT traffic management is currently a very hot topic. PSNC, a research organization and operator of a metropolitan network as well as a national education network, together with NoviFlow Inc., a leading provider of high-performance OpenFlow-based switching solutions, and Spirent, a provider of testing solutions, have prepared a proof-of-concept IoT application for Smart Cities and Smart Buildings. The demonstration shows the advantages of SDN in IoT applications for owners of smart spaces where multiple tenants can independently deploy their own sets of smart things over a common network and cloud infrastructure. The proposed solution resolves the problem of how to deploy and handle IoT traffic for multiple parties via a common, centrally managed SDN-based network infrastructure. One of the biggest challenges facing networks in the coming years is the tremendous growth in the amount of traffic which will be generated by IoT devices. In this context, SDN's ability to adjust network flows and services both dynamically and cost-effectively seems to be promising.

II. Architecture

In the demonstration, two separate IoT ecosystems are interconnected to the cloud via the SDN infrastructure.

The following equipment is used for the demo:

- Libelium Meshlium IoT system with two sets of sensors: Wasp Mote SmartCity and Ambient,
- Three NoviFlow NoviSwitch 2128 OpenFlow switches,
- Spirent TestCenter (STC) N4U with 4x10G card,
- HP ProLiant DL380 Gen 9 server with KVM and VMs for i) Ryu OpenFlow Controller, ii) IoT Application, iii) LXC containers.

The demonstration architecture is depicted in Fig. 1.

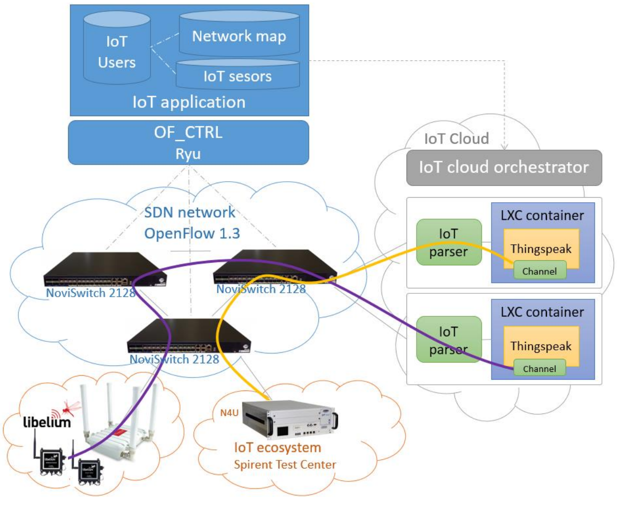

Fig. 1. Demonstration architecture

The solution's central logic is provided by an SDN-based IoT network application that resides on top of the SDN controller. It dynamically creates and manages end-to-end communication channels from IoT devices to the cloud using

2379-0369/17 $31.00 © 2017 IEEE
DOI 10.1109/EWSDN.2016.17

the SDN infrastructure. The application orchestrates cloud resources and manages IoT traffic between gateways and the cloud services by leveraging the TCP/UDP payload matching and handling features of NoviFlow NoviWare [1] for traffic analysis (IoT device recognition, categorization and policy enforcement), making it possible to automatically deploy IoT services over on-demand created LXC light containers for real-time data collection, storage, visualization and analysis.

The Libelium Meshlium system used for the demonstration comprises two sets of sensors, SmartCity and Ambient, each set having four different external sensors. Libelium's SmartCity set measures temperature, noise, ultrasound and humidity. Ambient's set of sensors measures humidity, luminosity and illumination. Moreover, Libelium's Meshlium IoT system collects the time and date of measurements and battery charge level of the sensors (which are powered from solar panels). The IoT sensors transmit data towards the gateway using the 802.15.4 wireless standard. The IoT gateways connect to the edge OpenFlow switches via Gigabit Ethernet links.

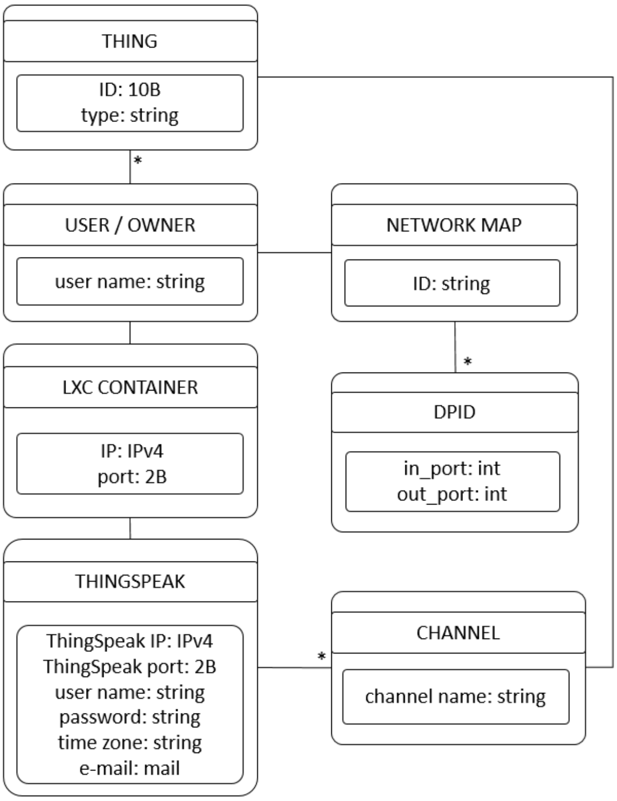

Fig. 2. Data model of the IoT application configuration (UML format)

Spirent TestCenter (STC) [2] is used to generate multi-device IoT test traffic in order to more realistically emulate a metro scale network. For the demonstration several traffic patterns are configured in the STC which are analogous to the one provided by Libelium Meshlium system, but configured to send random data.

ThingSpeak [3] is an open source software written in Ruby used in the demonstration for data collection, storage and visualization. It also provides tools for data analysis. This service is automatically deployed inside LXC containers using a cloud orchestrator Rest API. This interface to the cloud data center exposes methods for container management. For each tenant one ThingSpeak service containing multiple channels corresponding to that tenant's "things" is deployed. LXC containers are used on the cloud side because they are fast, light and work offline. They also isolate the environments and data of tenants from each-other as each tenant can access only their own ThingSpeak service.

The IoT cloud orchestrator and the IoT application running on top of the Ryu OpenFlow controller were designed and implemented from scratch in Python. Fig. 2 outlines the data model of the configuration file.

Currently there is no external GUI for configuration of the IoT application. The whole configuration is stored in a json format file.

III. Demonstration workflow

Libelium's gateway sends information from sensors every 3 minutes. This default behavior can be reprogrammed. The gateway may also store data in an external database, but a raw UDP stream is used in this demo. Data from the sensors is sent in Text ASCII format encapsulated into UDP.

To enable UDP transmission it is necessary to run a python script in the gateway: *python libelium_meshoul.demo.py.* The GUI of the STC is used to start transmission of test data from the Spirent traffic generator.

The SDN network with the Ryu controller works in the reactive mode. Any new UDP packet which does not match on existing rules installed in the edge switches (NoviSwitch OpenFlow switches) is sent to the controller (packet_in) and is handled by the IoT network application. Every first 10 bytes of the UDP payload is the ID of the "thing". The application parses the received packet and gathers information about the ID of the sensor. Then based on the ID and the configuration file it sets up a channel from the "thing" to the ThingSpeak service in the cloud. To create this channel and establish an end-to-end path the following steps are executed:

1) LXC container creation
2) ThingSpeak service deployment
3) IoT Parser object creation
4) SDN network provisioning

The IoT Application uses the REST API interface of the cloud orchestrator to manage cloud resources. First a new LXC container is created. Then a ThingSpeak service is deployed inside of it (this the most time consuming part of the process as ThingSpeak is implemented in Ruby and has not been optimized for performance) and a dedicated channel is created for data from the sensor (or set of sensors). The ThingSpeak portal is accessible via web page through the IP address and port configured in the application. Example data from the ThingSpeak channel is shown in Fig. 3.

In the next step the IoT application creates the IoT Parser object. It exposes the UDP socket for new UDP traffic in the cloud. Moreover, the IoT Parser uses the ThingSpeak API to put received and parsed data to the service.

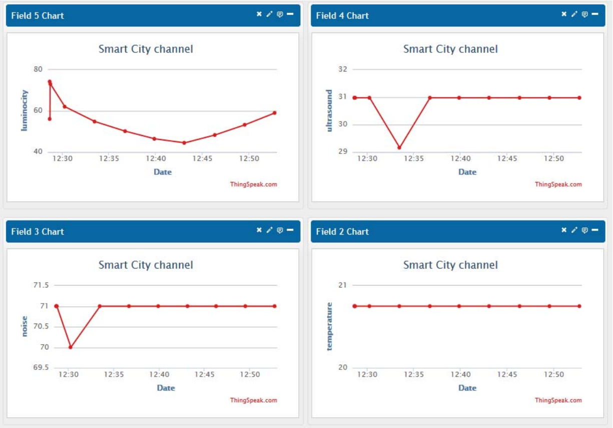

Fig. 3. Example charts from the ThingSpeak channel

Then, based on the network map, the IoT application provisions the path in the SDN network and configures all OpenFlow switches at once. It installs rules in the flow tables using the OpenFlow 1.3 protocol and two OpenFlow Experimenter extensions implemented by NoviFlow:

- matching on the first ten bytes of the UDP packet payload (ID of the thing),
- modifying the UDP packet by adding some useful metadata at the beginning of the UDP payload. An extra 25 bytes are added with information about i) the type of the IoT packet (useful for the parser in the cloud), ii) the API key of the ThingSpeak service (to put data from IoT Parser to ThingSpeak) and iii) the UDP socket port at the receiver side (in the container).

The new OpenFlow entry inside the flow table of the NoviSwitch matches on in_port to the thing ID, and specifies actions to: i) modify the UDP packet and ii) send it to the out_port.

It takes about 30 seconds in our test-bed from the moment when new UDP stream is recognized on the edge switch to the point when a dedicated ThingSpeak service is running inside LXC and the network from the “thing” to the ThingSpeak instance is established. The whole process is automated by the IoT application based on defined policies and configuration.

IV. SMART CITY USE CASE

The concept described in this paper may be implemented by any smart city where different sensors, cameras etc. could be managed by a unified application.

Our use case proposal is structured for Poznań SmartCity.

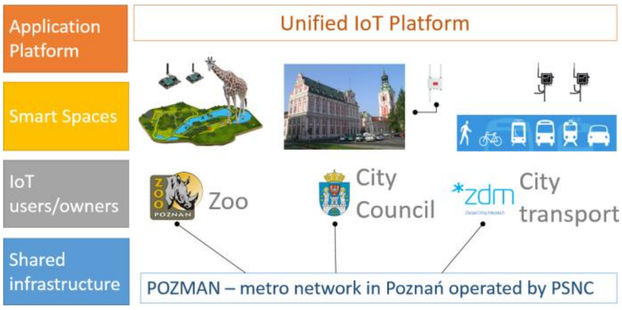

Fig. 4. Poznań Smart City use case

Poznań city is sliced (divided) into intelligent spaces (Smart Spaces) where “things” are managed by the unified platform for multiple tenants. All these spaces share network and cloud resources, operated by PSNC. In Fig. 4 three intelligent spaces are presented: Zoo, City Council and City Transport. Security and access to data from things can be managed according to network policy or delegated to individual tenants according to tenant service level agreements.

The solution described in this paper is based on the Ryu SDN controller and is a proof of concept limited to utilization of every single sensor by only one tenant (lack of virtualization). However, the application could easily be adapted to any popular orchestration system with a suitable API. The IoT application was tested with several sensors (Libelium and STC) and with the Spirent STC, but to evaluate performance and scalability of the whole system in a real smart city environment more tests are needed.

We are convinced more SDN-based Smart Cities concepts will appear in the near future in the context of evolving IoT technologies.

ACKNOWLEDGMENT

This work was conducted in close collaboration between research and commercial organizations (PSNC, NoviFlow, Spirent). Such involvement in close-concert of three research and development teams located in geographically disparate locations is very good example of the value of multi-party partnerships in defining and building useful products and solutions.

REFERENCES

[1] http://noviflow.com/products/noviware/, NoviWare, online, available 13.07.2015

[2] http://www.spirent.com/Products/TestCenter/DataCenter/OpenFlow, STC, online, available 13.07.2015

[3] https://thingspeak.com/, ThingSpeak, online, available 13.07.2015